

Island Name Matcher



Submitted By: Refractions Research Inc.
Suite 400 - 1207 Douglas Street
Victoria, BC, V8W-2E7
Phone: (250) 383-3022
Fax: (250) 383-2140

Table of Contents

1	INTRODUCTION.....	3
2	ISLAND NAME MATCHER.....	3
2.1	INPUTS.....	4
2.2	OUTPUTS.....	5
3	MATCHING ALGORITHM.....	6
3.1	DEFINITIONS	6
3.2	ISLAND MATCHING	7
3.2.1	<i>Matching Algorithm</i>	<i>7</i>
3.2.2	<i>Automated Matching Process</i>	<i>8</i>

Figures

Figure 1: Island Name Matcher Processes	3
Figure 2: Process Flow	10

Tables

Table 1: Unnamed Islands Shapefile	4
Table 2: Toponymy Shapefile	4
Table 3: Named Islands Shapefile.....	5
Table 4: List of Island Splits.....	5
Table 5: List of Issues	5
Table 6: Definitions.....	6

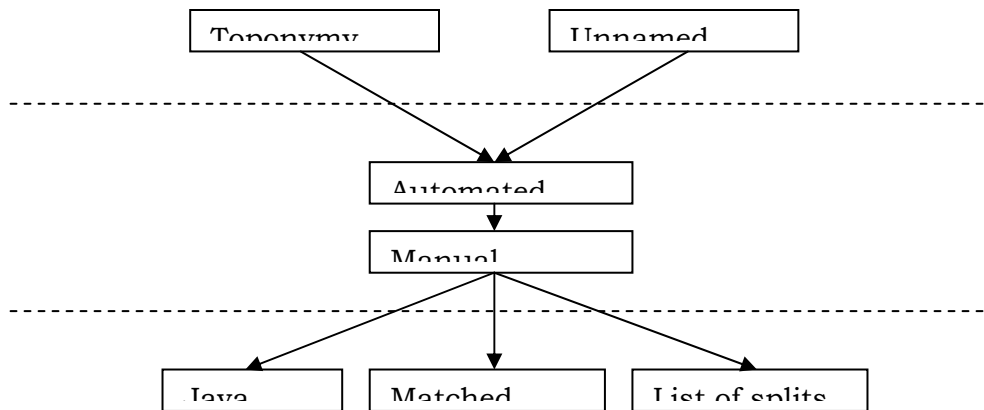
1 INTRODUCTION

The Island Name Matcher (INM) tool allows a user to match a set of island toponyms to set of island features. This tool includes an automated matching processes as well as a manual verification process where a user can verify and modify matches made during the automated matching process.

2 ISLAND NAME MATCHER

The following figure shows the stages for the Island Name Matcher process.

Figure 1: Island Name Matcher Processes



2.1 Inputs

The following tables describe the input files.

Table 1: Unnamed Islands Shapefile

Field	Description
Shape	Polygon
Island Id	Unique id attached to each island polygon for identification purposes.
Island Type	0 – Saltwater Island 1 – Freshwater Island 2 – Saltwater & Freshwater Island (Example: Swishwashi Island)
Contained	Id of the island polygon that contains this polygon. Contained will be –1 if this island polygon is not contained within another island polygon. [This does not include the bc mainland polygon].

Table 2: Toponymy Shapefile

Field	Description
Point	Point (location of a letter)
TopoId	Toponym ID (all letters with the same toponym id form a toponym)
Elem_id	Position of letter in toponym name.
Letter	Letter associated with point.
Size	Size of the text.
Rotation	Rotation of the text.
GnisId	ID to link toponyms belonging to the same feature together.
Code	Code of the toponym: 2300 – Island 2301 – Islet 2307 – Island Group 2308 – Islet Group 2319 – Group
Code_Cat	Category classification of toponym.

2.2 Outputs

The following tables describe the output files.

Table 3: Named Islands Shapefile

Field	Description
Shape	Polygon
Type	"Island"
Island_Id	Unique identifier to represent an island.
Name	Name matched to island.
Gnisid	GnisId matched to island.
Multinamed	1 if island is matched to more than one name. 0 if the island is matched to only one island.
QAQC	QAQC flags associated with the match.
Group_QAQC	Group QAQC flags associated with the match.

List of Splits

A comma-delimited file (.csv) will be produced of all the toponyms matched as an island split.

Table 4: List of Island Splits

Field	Description
Toponym (Name)	Name of toponym to be split.
GNIS ID	Gnis ID for toponym to be split.
TOPO ID	Topo ID of toponym to be split.
COMMENTS	Any comments entered by user during verification stage.
DESCRIPTION	Field informing the user the toponym has been flagged as a split.

List of Issues:

A comma-delimited file (.csv) will be produced of all the toponyms flagged with an issue by the user during the verification stage.

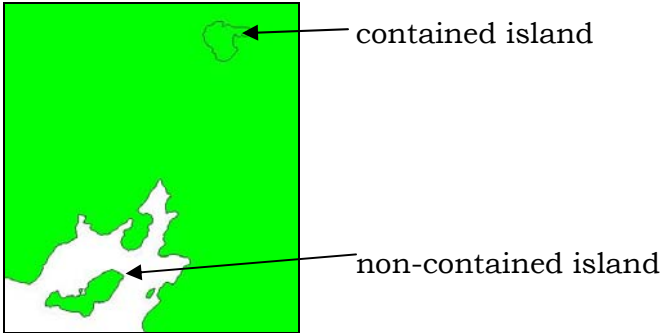
Table 5: List of Issues

Field	Description
Toponym (Name)	Name of toponym to be split.
GNIS ID	Gnis ID for toponym to be split.
TOPO ID	Topo ID of toponym to be split.
COMMENTS	Comments about the issue entered by the user during the verification stage.

3 MATCHING ALGORITHM

3.1 Definitions

Table 6: Definitions

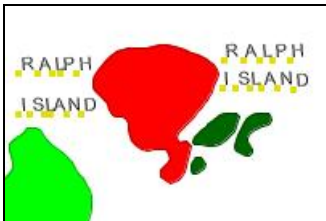
Toponym	A name on a map representing a feature. (<i>Example:</i> “Vancouver Island”)
TopoID	An island feature can have more than one toponym. Each toponym is assigned a unique ID. (<i>Example:</i> “Vancouver Island” may appear on a map 10 times. Each of these toponyms would be assigned a separate, unique TopoID.)
GnisID	All toponyms that represent the same feature are assigned the same GnisID. (<i>Example:</i> The 10 “Vancouver Islands” would all be assigned the same GnisID.)
Contained Island	An island is contained if it exists inside another island (within a river, lake, or other water feature). <i>Example:</i> An island that is inside a lake on Vancouver Island would be a contained island. An island inside of a lake on the bc mainland would not be a contained island. 

3.2 Island Matching

3.2.1 Matching Algorithm

The matching algorithm contains the following steps:

1. Load toponym and island datasets from the corresponding shapefiles.
2. Create toponyms from raw dataset. The toponym shapefile contains one line per letter per topo id. At this stage the letters are combined (by using the topo id) to generate a single toponym including all the letters associated with the topo id.
3. Perform the automated matching process.
4. Group toponyms based on GnisIDs. At this stage the toponyms and their matches are grouped together based on the GnisIDs.
 - For *group* toponyms: each island from every toponym match is added to the grouped toponym's match.
 - For *single* toponyms: the island with the highest rating from the previous step is selected as a match for the grouped toponym.



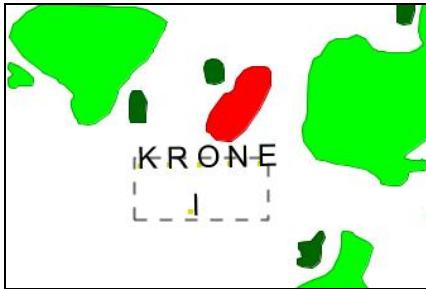
In this example there are two “Ralph Island” toponyms (with the same GnisID). At this stage these two toponyms (and their matches) are joined to form a single grouped toponym. This is done so that toponyms with the same GnisID cannot be matched to different islands.

5. At this stage the automated matching is finished. The islands, grouped toponyms, and matches are displayed to the user for verification and modification as necessary.

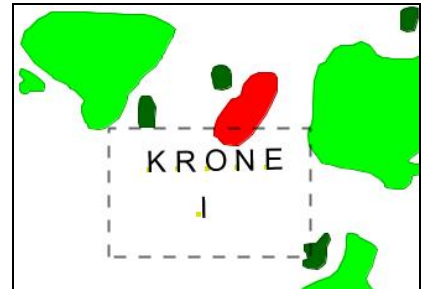
3.2.2 Automated Matching Process

The following steps outline the automated matching process:

1. Find all possible matches for each toponym. A possible match is determined by finding all islands within a given distance from the toponym. If no non-contained islands are found then the distance searched is increased until at least one non-contained island is found.



The initial search area for the toponym (left) turns up no possible island matches; therefore the search area is expanded (right). The expanded search area finds three possible island matches.



2. The best match for each toponym is found based on the possible matches.

For *GROUP* toponyms (as defined by the CODE field in input file):

The final match is set to include all the possible matches except contained islands.

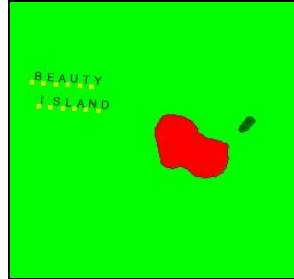


The possible matches found both islands (and neither are contained), therefore both islands are added to the final match for the toponym "Twin Islands".

For *SINGLE* toponyms (as defined by the CODE field in input file):

- i. Each non-contained island toponym is rated based on the following factors
 - area of island
 - closest distance from toponym to island
 - if the island has already been matched to another toponym with the same name.
 - if part of the toponym is contained inside the island. (In the "KRONE I" example above no part of the toponym is inside an island, but in the second example "TWIN ISLANDS" the "S" is contained in the second island).

- ii. The non-contained island with the best rating is found.
- iii. If the area of the toponym is less than 1% of the area of the island found in step (ii), then we search the contained islands (from the list possible matches). The largest contained island that is 'close' to the toponym is found.



In this example Beauty Island is contained inside Vancouver Island. If we didn't do this check then "Beauty Island" would automatically be matched to the Vancouver Island polygon, which is the incorrect match. Instead, "Beauty Island" is matched to the red contained island polygon.

- iv. Set the final match to the best non-contained island, unless a contained island was found in step (iii).

Figure 2: Process Flow

